

# PANDAS



## EL PARADIGMA

### 'SPLIT - APPLY - COMBINE'

ESTE PARADIGMA ES LA HERRAMIENTA MAS PODEROSA PARA EL ANALISIS DE DATOS A PARTIR DE UN DATA FRAME.

#### SPLIT

ESTA OPERACION DIVIDE AL DATAFRAME EN GRUPOS DE ACUERDO A UN CIEGTO CONJUNTO DE COLUMNAS, PARA ESTO UTILIZAMOS **GROUP BY**.

#### APPLY

CONSISTE EN APLICAR UNA FUNCION A CADA GRUPO. ES COMUN AGREGAR FUNCIONES DE AGREGACION PARA CADA GRUPO. SE PUEDEN APLICAR UNA O VARIAS FUNCIONES ESTADISTICAS O TRANSFORMACIONES.

#### COMBINE

ESTE PASO CONSISTE EN UNIR LOS GRUPOS NUEVAMENTE CON LO CUAL SE GENERA UN NUEVO DATAFRAME.

ENTONCES PODEMOS TENER:

- SPLIT - APPLY - COMBINE  
(AGREGACION)
- SPLIT - APPLY - COMBINE  
(FILTRADO)
- SPLIT - APPLY - COMBINE.  
(TRANSFORMACION)

COMENZANDO A UTILIZAR PANDAS LO PRIMERO  
ES IMPORTAR EL CSV.

### • IMPORT PANDAS AS PD

NOMBRE-DF = PD. READ-CSV ('ARCHIVO.CSV',  
ENCODING = 'LATIN-1')



ACA SE PUEDE AGREGAR  
USECOLS = ['COL1', 'COL3']  
ESTO PERMITE QUEDARTE  
SOLO CON ESAS COLUMNAS.

## SERIE

ES UNA SOLA DIMENSIÓN DEL DF, ES  
DECIR, UNA SOLA COLUMNA.

- DATA FRAME [ COLUMNA ]
- DATAFRAME . COLUMNA

A ESTAS SERIES PODEMOS APLICAR VARIOS METODOS

• VALUE-COUNTS() CUENTA LAS OCURRNCIAS DE  
CADA DATO.

ACA DENTRO PODEMOS COLOCAR  
NORMALIZE = TRUE Y CUENTA LAS  
OCURRNCIAS SOBRE EL TOTAL.

• SHAPE() SE APLICA A TODO EL DF Y DEVUELVE  
( FILAS, COLUMNAS ) SU CANTIDAD

• COUNT() CUENTA CUANTOS VALORES EN UNA SERIE  
SON DISTINTOS DE NAN.

• ISNULL() DEVUELVE UNA NUEVA SERIE INDICANDO  
TRUE SI ES NULL Y FALSE CASO CONTRARIO

- **HASNAN()** DEVUELVE TRUE O FALSE SI HAY<sup>2</sup>  
O NO AL MENOS UN VALOR NAN.
- **FILLNAN('NOMBRE')** PERMITE DEMPLAZAR  
NUMERO LOS NAN CON LO QUE  
QUIERAS.
- **DROPNA()** ELIMINA TODA LA FILA SI EN LA SEDE  
HAY UN NAN. SE PUEDE HACER  
**ESTADISTICAS** (INPLACE = TRUE)

- TENEMOS PARA APLICABLE A TODA LA COLUMNA
- MIN(), MAX(), MEAN(), MEDIAN(), STD()
- SUM(). ADEMÁS PODEMOS MULTIPLICAR, RESTAR  
O DIVIDIR SOBRE ESTAS ESTADISTICAS.
- MODE()

## LIMPIEZA DEL DF.

- DF-LIMPIO = DF[['COL1','COL5','COL3']]  
ME PERMITE GENERAR UN NUEVO DATA FRAME  
CON LAS COLUMNAS QUE ME INTERESAN EN EL  
ORDEN QUE YO QUIERO.
- INICIALMENTE EL DF TIENE COMO INDICE LA  
NUMERACIÓN 0,1...N, PERO SE PUEDE UTILIZAR  
COMO INDICE ALGUNA COLUMNA, ENTONCES:  
DF. SET-INDEX(['COL1'])  
• SET-INDEX(['COL1','COL2'])  
Y ESTO A SU VEZ PUEDE QUITARSE Y VOLVERA  
COMO ERA ANTES.  
DF. RESET-INDEX()

- **RENAME** (INDEX = ..., COLUMNS = '---')

EN EL CASO DE ←  
TENER INDEX

→ PUEDE HACERSE  
LOS MISMO PARA  
CAMBIAR EL  
NOMBRE.

{'NOMBRE-VIEJO': 'NOMBRE'  
                          NUEVO'}

- PUEDO CREAR UNA **NUEVA COLUMNA** ASIGNANDOLE LO QUE QUIERA.

DF['NOMBRE DE  
COL NUEVA'] =

- DF[COL1] > 10  
CREA UNA COL CON TRUE  
O FALSES
- DF[COL3]\*5
- = 0 O A UN NUMERO  
CUALQUIERA DIRECTO

- **ELIMINAR UNA COLUMNA** DEL DF

• **DROP** (COLUMNS = 'COL1', INPLACE = TRUE)

PD.

- **TO-DATETIME** (DF[COL]) CONVIERTETE EN FORMATO FECHA UNA COLUMNA PARA DESPACEDER AL DIA MES Y AÑO.

• DT. YEAR

• DT. MONTH

- PARA PODER **FILTRAR** CIERTAS COLUMNAS QUE NO NOS INTERESAN :

FILTRO = DF[COLUMNAS] == 80

DF-FILTADO = DF[FILTRO]

DE ESTA MANERA PODEMOS ELIMINAR LAS FILAS QUE NO CUMPLEN CON EL FILTRO.

- **NLABGEST (N)** DE VUELVE LOS N MAS GRANDES APLICADO A UNA COLUMNA.

- SORT-VALUES('COL' ASCENDING = TRUE)  
ORDENA LAS FILAS DE MAYOR A MENOR O  
VISEVERSA SEGUN LA COLUMNA.

## GROUP BY

- GROUPBY(['COL'])  
NOS PERMITE AGRUPAR  
POR LOS VALORES EN  
ESA COLUMNA  
ASI SOLO NO HACE NADA
- NUEVO = DF.GROUPBY(['COL'])  
• AGG({'QUE COL': [FUNCIONES]})  
ESTO QUEDA CON INDICES DE 2 NIVELES PARA  
LLEVARLO AL FORMATO NORMAL LE APLICO UN  
• RESET-INDEX().
- \* RECORRER QUE AVECES SE PUEDE AGRUPAR  
POR DOS COLUMNAS (O MAS).

• AGG  
ESTO SE PUEDE  
APLICAR A VARIAS  
COLUMNAS.

## TRANSFORM

EL TRANSFORM LE APLICA UNA FUNCION A LOS GRUPOS  
GENERADOS POR GROUPBY() Y DEVUELVE UNA COL  
DEL MISMO LARGO DEL SET, DONDE CADA ENTRADA  
TIENE LA FUNCION APLICADA.

TRAIN[NUEVA-COL] = TRAIN.GROUPBY('KEYWORD')  
['TARGET'].TRANSFORM(  
PUEDE SER CUALQUIER LAMBDA → 'COUNT')

- **APPLY()** PERMITE APLICAR UNA FUNCIÓN A UNA COLUMNA DEL DF.

## **COMBINACIONES**

- **APPEND()** UNE DOS DFS SIN TENER EN CUENTA NINGUN INDEX NADA.
- **CONCAT()**
- **MERGE (DF2, HOW = 'MODO')** PERMITE UNION DOS DFS DE DISTINTAS MANERAS LEFT, RIGHT, INNER, OUTER.